Σημασιολογικός Ιστός και Ευφυείς Εφαρμογές

(*2η Εργασία / OWL*)

Μέλη Ομάδας:

- Σωτήριος Λουκάς Καμπύλης, ΑΕΜ: 3805

- Ραφαήλ Τανακίδης, ΑΕΜ: 3814

Η συγκεκριμένη εργασία υλοποιήθηκε από τα 2 παραπάνω μέλη και αναπαριστά την μοντελοποίηση ενός κοινού θεματικού πεδίου ενδιαφέροντος, δηλαδή μιας **Αλυσίδας Βιβλιοθήκης-Βιβλιοπωλείου.** Η εργασία υλοποιήθηκε εξίσου και από τα 2 μέλη, καθώς βρισκόμασταν σε επικοινωνία από την αρχή της υλοποίησης της μέχρι το τέλος. Η 2η εργασία βασίζεται στην οντολογία που αναπτύχθηκε στα πλαίσια της 1ης εργασίας και αναγκαστικά θα ξαναπαρουσιαστούν οι κλάσεις, ιδιότητες κτλ. (θα δοθεί έμφαση σε αυτές που επεξεργάστηκαν έτσι ώστε να ικανοποιούνται οι απαιτήσεις της 2ης εργασίας). Συγκεκριμένα, πραγματοποιήσαμε αλλαγές σε κάποιες κλάσεις ώστε να ενσωματώσουμε περιορισμούς κλάσεων και χρησιμοποιήσαμε την τομή κλάσεων ώστε να ορίσουμε μία νέες κλάσεις ενώ μέσω των περιορισμών υλοποιήσαμε ισοδυναμίες κλάσεων. Μάλιστα, φτιάξαμε (ή αλλάξαμε) συμμετρικές, μεταβατικές, αντίστροφες και συναρτησιακές ιδιότητες. Επίσης, δημιουργήσαμε πολλά νέα αντικείμενα (κυρίως για να μπορέσουμε να κάνουμε πολλές δοκιμές με τον ***Reasoner –*** *στην περίπτωση μας ο* **HermiT**!). Παρακάτω θα αναλύσουμε περαιτέρω την

σημασία των κλάσεων/ιδιοτήτων για το μοντέλο μας (έτσι ώστε να γίνει καλύτερα αντιληπτό). Ωστόσο, δεν θα αναλύσουμε εξονυχιστικά όλο το μοντέλο γιατί χρησιμοποιήσαμε πάρα πολλά πράγματα και θα μας έπαιρνε τουλάχιστον 10 σελίδες (μόνο για αυτό), οπότε θα αναλύσουμε ένα μεγάλο δείγμα από το μοντέλο μας.

ΚΛΑΣΕΙΣ/ΥΠΟ-ΚΛΑΣΕΙΣ

Κλάσεις και Υπό-κλάσεις που χρησιμοποιήθηκαν (με την σειρά που εμφανίζονται στο Protégé): 1) Action, 2) Borrow, 3) Buy, 4) Read, 5) BorrowingCard, 6) ClubJoin, 7) Clubs, 8) Communication, 9) Donation, 10) HasBorrowingCard, 11) Location, 12) Organization, 13) Person, 14) Author, 15) Famous_Author, 16) Client, 17) Adult, 18) Stuff, 19) Manager, 20) Night_Worker, 21) Product, 22) Book, 23) Cheap_Book, 24) Expensive_Book, 25) Programs, 26) AfterSchoolHomeWorkHelp, 27) Games, 28) Publisher, 29) Famous_Publisher 30) Rank, 31) Receipt, 32) Salary, 33) Schedule, 34) Security, 35) Camera, 36) Guard, 37) Store, 38) Supplier, 39) Technology, 40) Pc, 41) Printer & 42) Warehouse.

Σύμφωνα με το Protégé πάνω στις κλάσεις φτιάξαμε τα *Restrictions* που αναφέρονται στην εκφώνηση (necessary **>10** & sufficient **>10**). Παρακάτω αναφέρονται ονομαστικά (μερικά από αυτά):

1) Για την κλάση Clubs: necessary restriction -> *Club Type some xsd:string, (όπου σημαίνει ότι κάποιο Club πρέπει να έχει 1 ή περισσότερα Club Type το οποίο παίρνει τιμή συμβολοσειρά).*
2) Για την κλάση Person: 2 necessaries restrictions -> *PersonAge only xsd:integer, (όπου σημαίνει ότι ένα Person πρέπει να έχει μία και μόνο μία ηλικία, η οποία θα είναι ένας ακέραιος αριθμός) & PersonName only xsd:integer, (όπου σημαίνει ότι το Person πρέπει να έχει ένα και μόνο ένα όνομα, το οποίο θα είναι μία συμβολοσειρά).*

3) Για την κλάση HasBorrowingCard: necessary restriction - > HasBorrowingCardClient only Client, (*όπου σημαίνει ότι ένας πελάτης μπορεί να έχει μόνο μία δανειστική κάρτα*).
4) Για την κλάση NightWorker: sufficient restriction -> Stuff and (ScheduleTime value “Night”), *(όπου σημαίνει ότι ένα Stuff που δουλεύει σε νυχτερινό ωράριο είναι αυτόματα και νυχτοφύλακας).*

(τα υπόλοιπα βρίσκονται μέσα στο .owl αρχείο και ακολουθούν την ίδια λογική με τα παραπάνω!)

Για το *unionOf*: Βρίσκονται μέσα στις 2 κλάσεις Programs (AfterSchoolHomeWorkHelp + Games) & Technology (Pc + Printer). Η «έξω κλάση» αποτελεί την ένωση των 2 «μέσα κλάσεων».

Για το *intersectionOf*: Βρίσκονται μέσα στις 4 κλάσεις Famous_Author, Adult, Manager & Night_Worker. Για παράδειγμα για να είναι ένα αντικείμενο Famous_Author πρέπει ταυτόχρονα να είναι και Author και να έχει γράψει πάνω από 3 βιβλία. Ενώ για να είναι ένα αντικείμενο Manager, πρέπει να είναι Stuff και ταυτόχρονα να έχει τιμή το Property Manager_Is αυτού του αντικειμένου. Το οπόιο για να αποκτήσει τιμή πρέπει το αντικείμενο να έχει στο Object Property HasRank την τιμή Rank_4 που σημαίνει Manager.

Όσο αφορά τα disjointWith: Αυτό γίνεται στις κλάσεις: Action-ClubJoin, Clubs-Organization, Communication-Location, Donation-HasBorrowingCard κτλ. Το οποίο μας ενημερώνει ότι μεταξύ αυτών των 2 κλάσεων δεν υπάρχει καμία σχέση που να τις ενώνει.

ΙΔΙΟΤΗΤΕΣ/ΥΠΟ-ΙΔΙΟΤΗΤΕΣ

Ιδιότητες και Υπό-ιδιότητες που χρησιμοποιήθηκαν (με την σειρά που εμφανίζονται στο Protégé): 1) ActionClient, 2)

BookAuthor, 3) BookCategory, 4) BookPublisher, 5) BookReleaseDate, 6) BookTitle, 7) BorrowBook, 8) BorrowEnd, 9) BorrowingCardId, 10) BorrowingCardUntil, 11) BorrowStart, 12) BorrowStuff, 13) BuyProduct, 14) BuyReceipt, 15) BuyStuff, 16) CameraId, 17) ClubJoinClub, 18) ClubJoinPerson, 19) ClubType, 20) CommunicationEmail, 21) CommunicationPhone, 22) DonationAmount, 23) DonationPerson, 24) GuardIs, 25) HasBorrowingCardCard, 26) HasBorrowingCardClient, 27) LocationAddress, 28) LocationCity, 29) LocationZipCode, 30) OrganizationArrive, 31) OrganizationContent, 32) OrganizationLeave, 33) PersonAge, 34) PersonId, 35) PersonName, 36) ProductName, 37) ProductValue, 38) PublisherId, 39) PublisherLocation, 40) PublisherName, 41) ReadBook, 42) SalaryAmount, 43) SalaryStuff, 44) ScheduleStuff, 45) ScheduleTime, 46) StoreCommunication, 47) StoreLocation, 48) SupplierName, 49) SupplierType, 50) SupplierTypeQuantity, 51) TechnologyAvailable, 52) WarehouseCommunication & 53) WarehouseLocation +(μερικές ακόμα!).

Αρχικά, Το Inverse Of αναφέρεται στο αντίστροφο Property, από ένα άλλο. Μερικά από αυτά που χρησιμοποιήσαμε είναι τα BookAuthor-author_write, Has_Salary-SalaryStuff και BookPublisher-publish_book Για παράδειγμα το BookAuthor έχοντας domain Book και Range Author αναφέρει από ποιον συγγραφέα έχει γραφτεί το βιβλίο. Ενώ το Property, author_write, έχει domain Author και range Book και αναφέρεται σε ποιο βιβλίο έχει γράψει ο συγγραφέας.

Στην Συνέχεια, SymmetricProperty είναι π.χ. η *SameRank* επειδή έχει ίδιο domain και range, το οποίο σημαίνει ότι ισχύει και ανάποδα. Δηλαδή έστω a *SameRank* b, τότε ισχύει και b sameRank a.

Ύστερα, TransitiveProperty, δηλαδή μεταβατικότητα, το οποίο έχουμε ορίσει σε κάποια Property, ένα από αυτά είναι το property Branch_store_of που σημαίνει υποκατάστημα αλλού καταστήματος. Για παράδειγμα το store 4 είναι υποκατάστημα του store 3 και το store 3 είναι υποκατάστημα του store 2, με την μεταβατικότητα συμπεραίνουμε ότι το store 4 είναι υποκατάστημα του store 2.

Σχετικά με το FunctionalProperty έχουμε το Has_Salary που σημαίνει ότι συναρτησιακά ισχύει η σχέση 1 προς 1

(κάθε υπάλληλος αντιστοιχείται σε έναν μισθό). Το ίδιο ισχύει για BookPublisher.

Τέλος, το InverseFunctionalProperty έχουμε το publish Book όπου ένας εκδότης μπορεί να εκδόσει περισσότερα από ένα βιβλία.

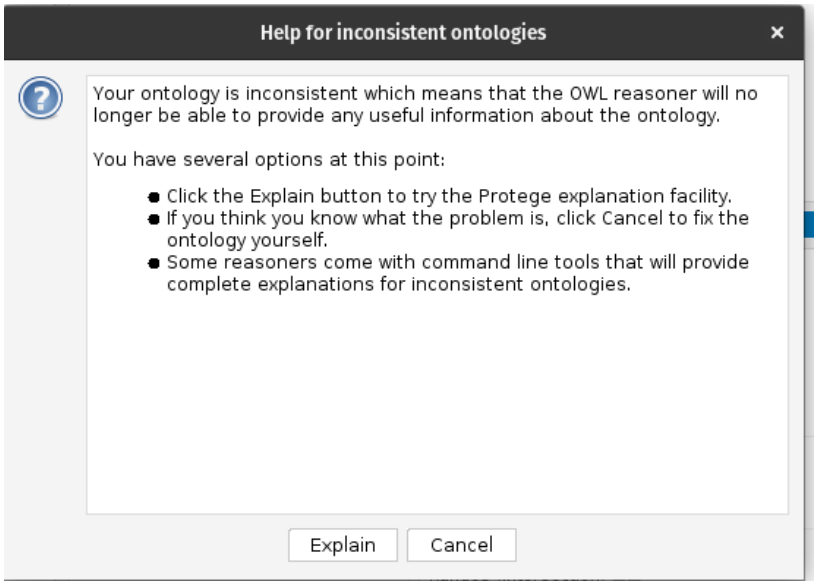
ΑΣΥΝΕΠΕΙΕΣ

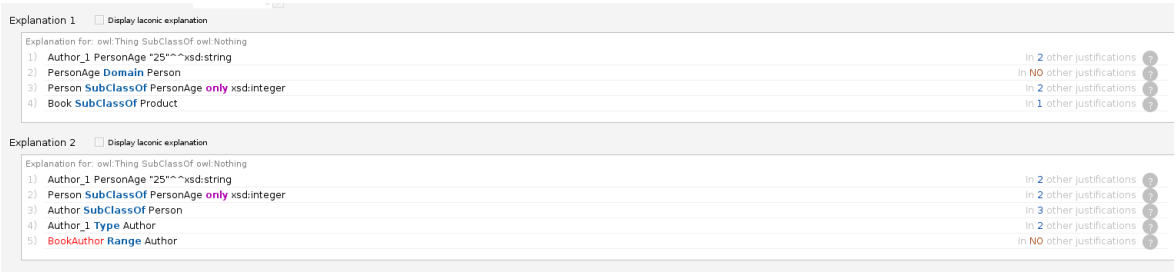
(**Σημείωση**: Το μοντέλο μας έχει διορθωθεί έτσι ώστε να μην βγάζει ασυνέπειες (ύστερα από εφαρμογή του *Reasoner*)!

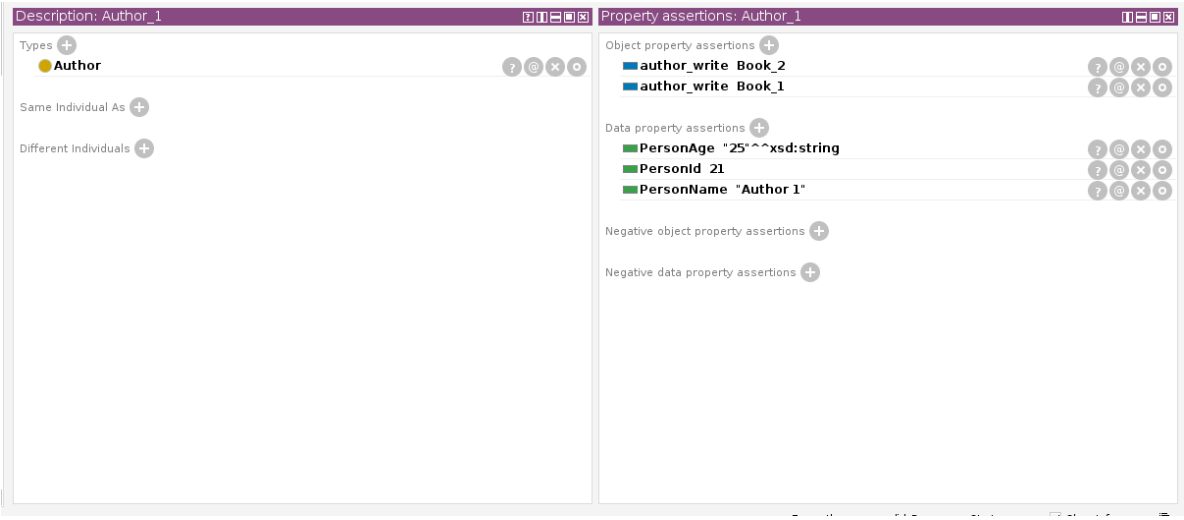
1) Μια ασυνέπεια που μπορούμε να βρούμε είναι στην περίπτωση που το Object Property BookAuthor είναι symmetric. Καθώς αυτό συνεπάγεται στο ότι ένα βιβλίο μπορεί να είναι και Author ενώ ένας συγγραφέας μπορεί να είναι και Book.

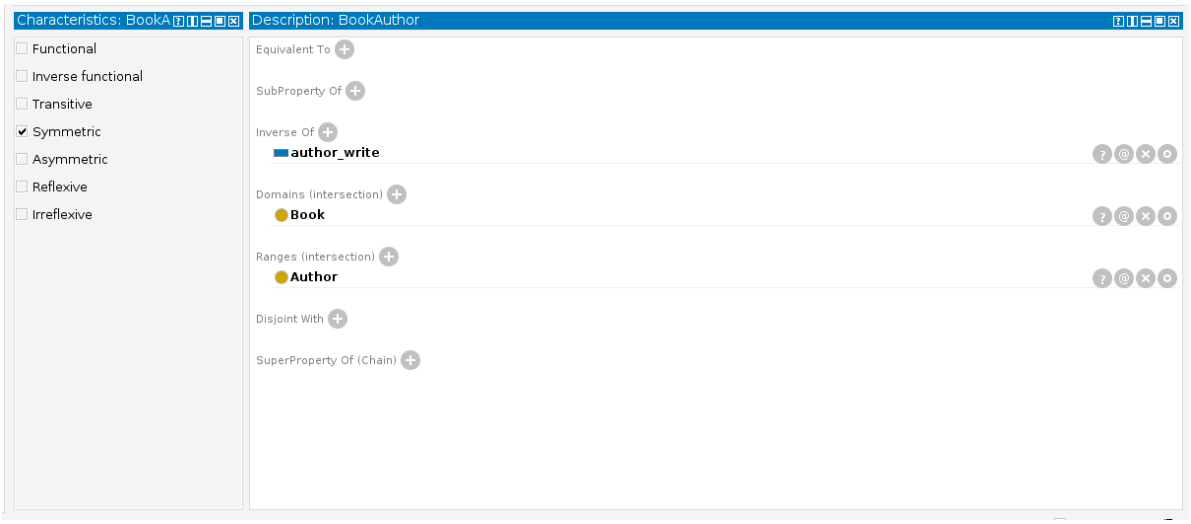
2) Μια άλλη ασυνέπεια που μπορούμε να βρούμε είναι στην περίπτωση που το Person είναι string ενώ κανονικά θα έπρεπε να είναι integer.

Οι παραπάνω ασυνέπειες παίρνονται από τον Reasoner και φαίνονται στα παρακάτω screenshots:


Help for inconsistent ontologies
Your ontology is inconsistent which means that the OWL reasoner will no longer be able to provide any useful information about the ontology.
You have several options at this point:
Click the Explain button to try the Protege explanation facility.
If you think you know what the problem is, click Cancel to fix the ontology yourself.
Some reasoners come with command line tools that will provide complete explanations for inconsistent ontologies.
Explain
Cancel


Explanation 1
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Author_1 PersonAge "25"^^xsd:string
2) PersonAge Domain Person
3) Person SubClassOf PersonAge only xsd:integer
4) Book SubClassOf Product
Explanation 2
Display laconic explanation
Explanation for: owl:Thing SubClassOf owl:Nothing
1) Author_1 PersonAge "25"^^xsd:string
2) Person SubClassOf PersonAge only xsd:integer
3) Author SubClassOf Person
4) Author_1 Type Author
5) BookAuthor Range Author


Description: Author_1
Types
Author
Same Individual As
Different Individuals
Property assertions: Author_1
Object property assertions
author_write Book_2
author_write Book_1
Data property assertions
PersonAge "25"^^xsd:string
PersonId 21
PersonName "Author 1"
Negative object property assertions
Negative data property assertions


Characteristics: BookA
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: BookAuthor
Equivalent To
SubProperty Of
Inverse Of
author_write
Domains (intersection)
Book
Ranges (intersection)
Author
Disjoint With
SuperProperty Of (Chain)